CLAIMS

~~Sub G4 B1~~ C1

1. A method of managing memory in a multi-threaded processing environment including respective local thread stacks and heaps and a global heap, said method comprising:

creating an object in a thread heap; and

monitoring whether the object is reachable from anywhere other than the thread stack.

2. A method as claimed in claim 1 further comprising:

associating a local status with the object;

changing the status of the object to global under certain conditions.

3. A method as claimed in claim 2 further comprising deleting from the thread heap one or more local objects when they are not reachable from a local root.

4. A method as claimed in claim 3 where reachability is determined by tracing from the local root.

5. A method as claimed in claim 4 wherein the status of an object in the thread heap is changed to global if the object is assigned to a static variable or if the object is assigned to a field in a global object.

6. A method as claimed in claim 3 further comprising intercepting assignment operations to an object in a thread heap to access whether the object status should be changed.

7. A method as claimed in claim 6 wherein the multithreaded processing environment is a virtual machine.

8. A method as claimed in claim 7 wherein the virtual machine comprises an interpreter and the write operation code in the interpreter is modified to perform the checking of assignment of the object.

9. A method as claimed in claim 8 wherein the virtual machine comprises a just in time compiler wherein native compiled write operation code includes native code to perform the checking of assignment of the object.

10. A method as claimed in claim 9 further comprising using spare capacity in the object header for the flag.

11. A method as claimed in claim 10 further comprising using multiples of 2 or more bytes in a thread heap to store the objects whereby there is at least one spare bit in the object length variable and using the at least one spare bit as the flag.

12. A method as claimed in claim 11 further comprising moving objects whose status is global from the thread heap to a global heap.

13. A method as claimed in claim 12 further comprising compacting the reachable local objects in a thread heap.

14. A method as claimed in claim 1 wherein certain objects are associated with a global status on creation.

15. A method as claimed in claim 14 where said certain objects include class objects.

16. A method as claimed in claim 14 further comprising the step of analysing whether an object is likely to be made global and associating such an object with a global status on creation.

17. A method as claimed in claim 16 further comprising allocating objects assigned as global on creation to the global heap.

18. A system for managing memory in a multi-threaded processing environment comprising:

respective local thread stacks and heaps;

a global heap;

means for creating an object in a thread heap; and

means for monitoring whether the object is reachable from outside the thread heap.

19. A system as claimed in claim 18 further comprising means for associating a local status with the object and means for changing the status of the object to global under certain conditions.

20. A system as claimed in claim 19 further comprising means for deleting from the thread heap one or more local objects when they are not reachable from a local root.

21. A system as claimed in claim 20 further comprising:
means for changing the status of an object in the thread heap to global if the object is assigned to a static variable or if the object is assigned to a field in a global object.

22. A computer program product stored on a computer readable storage medium for, when executed on a computer, performing a method of managing memory in a multi-threaded processing environment including respective local thread stacks and heaps and a global heap, said method comprising:
creating an object in a thread heap; and
monitoring whether the object is reachable from outside the thread heap.

23. A method as claimed in claim 22 further comprising:
associating a local status with the object;
changing the status of the object to global under certain conditions.

24. A method as claimed in claim 23 further comprising deleting from the thread heap one or more local objects when they are not reachable from a local root.

25. A method as claimed in claim 24 where reachability is determined by tracing from the local root.

26. A method as claimed in claim 25 wherein the status of an object in the thread heap is changed to global if the object is assigned to a static variable or if the object is assigned to a field in a global object.